Remarks

Claims 1-25 remain pending.

Claim Rejections—35 USC 103

Claims 1, 3-8, 15, and 17-18 stand rejected under 35 USC 103 as being unpatentable over Killian (US 6,760,888) in view of Colwell (US 4,833,599). Claims 9-14 and 19-24 stand rejected under 35 USC 103 as being unpatentable over Killian in view of Colwell and further in view of Raina. Claims 2, 16 and 25 stand rejected under 35 USC 103 as being unpatentable over Killian in view of Colwell and further in view of Murthi.

Applicants respectfully traverse these rejections.

Claims 1-14

Claim 1, as previously presented, recites as follows.

1. A method of compiling a program to be executed on a target microprocessor with multiple execution units of a same type, the method comprising:
   selecting, by a program compiler, one of the execution units for testing;
   scheduling, by the program compiler, execution of diagnostic code on the selected execution unit; and
   scheduling, by the program compiler, execution of program code on remaining execution units of the same type,
   wherein said execution of diagnostic code on the selected execution unit and said execution of program code on the remaining execution units are **scheduled to be performed in parallel.**

(Emphasis added.)

As seen above, claim 1 clearly claims a "method of compiling a program" where **each claimed step is performed "by a program compiler"**. As is known in the art, a program compiler **transforms source code to object code**. Moreover, claim 1 recites that **the selection of one of the execution units is for testing** and that **diagnostic code is scheduled on the selected execution unit**. As such, claim 1 relates to the use of a program compiler to test execution units in a microprocessor by scheduling diagnostic code to run on a selected execution unit while the **remaining execution units continue in parallel to execute program code.** Said execution of diagnostic code on the selected execution unit and said execution of program code on the remaining execution units are **scheduled to be performed in parallel.**

I. KILLION DOES NOT DISCLOSE ANY OF THE CLAIM ELEMENTS

Per page 3 of the latest office action, the Examiner states that Killion does not disclose any of the following limitations in claim 1.

> selecting, by a program compiler, one of the execution units for testing;
> scheduling, by the program compiler, execution of diagnostic code on the selected execution unit; and
> scheduling, by the program compiler, execution of program code on remaining execution units of the same type,
> wherein said execution of diagnostic code on the selected execution unit and said execution of program code on the remaining execution units are scheduled to be performed in parallel.

Applicants do not traverse this conclusion by the Examiner.

II. COL. 7, LINES 5-12 OF COLWELL DOES NOT DISCLOSE OR SUGGEST THE CLAIM LIMITATION OF "SELECTING, BY A PROGRAM COMPILER, ONE OF THE EXECUTION UNITS FOR TESTING"

Page 3 of the latest office action asserts col. 7, lines 5-12 of Colwell discloses the claim limitation of "selecting, by a program compiler, one of the execution units for testing." Applicants respectfully disagree with this assertion.

Col. 7, lines 5-12 of Colwell recites as follows.

> The compiler uses various methods to select the best of the multiple projected traces and calls upon a "disambiguator" to assist in creating code that has parallel structure. The disambiguator method decides whether or not implied memory references result in a program conflict, that is, whether or not memory references can be executed in parallel.

As seen from the above quotation, this citation in Colwell discloses selecting the best of multiple projected traces and calls upon a "disambiguator" to assist in creating code that has a parallel structure.

There is no disclosure in the above quotation regarding the claim limitation of "selecting, by a program compiler, one of the execution units for testing." In particular, the cited **disambiguator** method appears to be irrelevant to the limitation of **"selecting ... one of the execution units for testing**." <u>If the Examiner maintains this rejection, applicants respectfully request an explanation of how the cited text from Colwell discloses this claim limitation.</u>

III. COL. 2, LINES 36-58 AND COL. 3, LINES 15-20 OF COLWELL DO NOT DISCLOSE OR SUGGEST THE CLAIM LIMITATIONS OF "SCHEDULING, BY A PROGRAM COMPILER, EXECUTION OF DIAGNOSTIC CODE ON THE SELECTED EXECUTION UNIT" AND "SCHEDULING, BY THE PROGRAM COMPILER, EXECUTION OF PROGRAM CODE ON REMAINING EXECUTION UNITS OF THE SAME TYPE"

Page 4 of the latest office action asserts col. 2, lines 36-58 and col. 3, lines 15-20 of Colwell discloses the claim limitations of "scheduling, by the program compiler, execution of diagnostic code on the selected execution unit; and scheduling, by the program compiler, execution of program code on remaining execution units of the same type." Applicants respectfully disagree with this assertion.

Col. 2, lines 38-58 recites as follows.

> The invention features circuitry for processing a plurality of the branch instructions to be executed in parallel and having a hierarchical priority for denoting a tentative order of execution of the instructions, and each processing unit features circuitry for executing a selected plurality of the instructions during a single machine cycle, the **selected plurality of instructions including at most one branch instruction**, a first and a second branch bank register sets, circuitry employing the branch bank register sets for determining whether test conditions associated with a branch instruction are satisfied, circuitry for simultaneously and independently determining a target program counter address for the branch instruction to be executed by the arithmetic processor, circuitry responsive to other processors of the apparatus for determining whether the arithmetic processor is the processor having the highest priority branch instruction with a satisfied branch condition, and circuitry for setting a next instruction address equal to the determined target program counter address of the processor if it has the test condition satisfied highest priority branch instruction.
>
> (Emphasis added.)

In particular, the latest office action emphasizes the disclosure of Colwell that "selected plurality of instructions including at most one branch instruction...."

Col. 3, lines 15-20 of Colwell recites as follows.

> **Each processing cluster features, for executing a branch instruction, circuitry responsive to the instruction data directed thereto for testing**, during said execution, whether the branch instruction has a test condition associated therewith which is true....
>
> (Emphasis added.)

In particular, the latest office action emphasizes the disclosure of Colwell that "Each processing cluster features, for executing a branch instruction, circuitry responsive to the instruction data thereto for testing ...."

Both of the above citations to Colwell relate to selecting and executing a **branch** instruction. Applicants respectfully submit that, typically, a conditional branch instruction is an instruction in a computer program that may or may not be taken during program execution. If the branch is not taken, then the flow of control is unchanged and the immediately following instruction is executed. If the branch is taken, then the next instruction to be executed is at a different location in memory as indicated by the branch instruction. There are also unconditional branch instructions where the branch is always taken.

Applicants respectfully submit that selecting and executing a **branch** instruction is **unrelated and irrelevant** to the claim limitations of "scheduling, by the program compiler, **execution of diagnostic code on the selected execution unit**; and scheduling, by the program compiler, **execution of program code on remaining execution units of the same type.**" For example, applicants respectfully submit that the "**testing**" referred to in col. 3, lines 15-20 of Colwell relates to **test conditions that determine whether or not a branch is to be taken**. In contrast, the claim 1 requires selecting an execution unit for **diagnostic** testing.

<u>If the Examiner maintains the assertion that these citations regarding branch instructions somehow disclose these claim limitations, then Applicants respectfully request an explanation of how the selecting and executing branch instructions per Colwell is relevant to the claim limitations where diagnostic code is scheduled to be executed on a selected execution unit, and program code is scheduled to be executed on remaining execution units of a same type.</u>

For at least the above-discussed reasons, applicants respectfully submit that claim 1 is patentably distinguished over Killian in view of Colwell.

Claims 2-14 depend from claim 1. Hence, for at least the same reasons as discussed above in relation to claim 1, applicants respectfully submit that claims 2-14 are now also patentable.

Further regarding claims 2 and 3, applicants respectfully submit that the cited **BSP (bootstrap processor)** of Murthi does <u>not</u> teach or disclose the claimed limitation which pertains to **selection of a unit for diagnostic testing by a program compiler**. Hence, applicants respectfully submit that the citation to Murthi does **not** disclose or teach the limitations of claims 2 and 3. <u>If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the bootstrap processor is relevant to selection of a unit for diagnostic testing by a program compiler.</u>

Claim 4 is further distinguished over Killion and Colwell because the cited reference to col. 11, lines 25-26 of Colwell relates to **assigning priority levels for**

intra-processor unit data transfers. In contrast, claim 4 recites "**setting a level of aggressiveness for scheduling the testing of the execution units**." (Emphasis added.) Applicants respectfully submit that the disclosed priority levels for intra-processor data transfers are **irrelevant** to the limitation of claim 4. If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the intra-processor unit data transfers are relevant to aggressiveness levels for scheduling testing of execution units.

Claim 5 is further distinguished over Killion and Colwell because the cited reference to col. 22, lines 24-32 of Colwell relates to a **"multi-set" cache**. In contrast, claim 5 recites "**applying an aggressiveness-dependent algorithm to determine when to schedule all available units for execution of the program code and when to schedule parallel execution of the program code and the diagnostic code**." (Emphasis added.) Applicants respectfully submit that the disclosed multi-set cache is **irrelevant** to the limitation of claim 5. If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the multi-set cache is relevant to the claimed aggressiveness-dependent algorithm for scheduling program code and diagnostic code.

Claim 6 is further distinguished over Killion and Colwell because the cited reference to col. 11, lines 31-34 of Colwell relates to a two-bit field representing "the state of **bus usage** by the processor." (Emphasis added.) For example, the field can be "11 = no buses being used." In contrast, claim 6 requires that a lowest level of aggressiveness for scheduling the testing of the selected execution unit comprise **turning off said testing of the selected execution unit**. Applicants respectfully submit that the disclosed bus usage field is **irrelevant** to the limitation of claim 6. If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the bus usage field discloses the claimed lowest level of aggressiveness for scheduling the execution unit testing.

Further regarding claim 10, a portion (col. 3, lines 19-22) of Raina is cited which pertains to **the testing of multiple processor cores by comparing output signals if the input signals are the same**. In contrast, claim 10 requires that" the scheduled diagnostic code **performs diagnostic operations from a test pattern comprising operations with known expected results**." The cited portion of Raina does **not** teach or disclose such a test pattern with **known** expected results. On the contrary, applicants respectfully submit that **Raina performs its <u>comparison</u> operation precisely because the results are <u>unknown</u>**. <u>If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the cited portion of Raina discloses the limitation of claim 10.</u>

Further regarding claim 12, a portion (col. 3, lines 15-22) of Raina is cited which pertains to the **testing of multiple processor cores by comparing output signals if the input signals are the same** and where "... the test result is either a pass or a fail indication." In contrast, claim 12 requires that "the scheduled diagnostic code **jumps to a fault handler if the compared results are different**." (Emphasis added.) Applicants respectfully submit that **the cited portion of Raina does not teach or disclose such jumping to a fault handler.** <u>If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the cited portion of Raina discloses the limitation of claim 12.</u>

Further regarding claim 13, a portion (col. 3, lines 17-19) of Raina is cited that states "If all of the input signals do not have the same logic value, then <u>a disable state value is output</u>, at state 106." (Emphasis added.) Applicants respectfully submit that <u>this state disables the comparison of the output signals</u>. In contrast, claim 13 requirse "code to **remove a faulty execution unit from use**...." (Emphasis added.) Applicants respectfully submit that the disable state of Raina does <u>not</u> remove a faulty execution unit from use. <u>If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the cited portion of Raina discloses the limitation of claim 13.</u>

Further regarding claim 14, the portion (col. 3, lines 17-19) of Raina is again cited that states "If all of the input signals do not have the same logic value, then **a disable state value is output**, at state 106." (Emphasis added.) Applicants respectfully submit that **this state disables the comparison of the output signals**. In contrast, claim 14 requires "code to **perform a system halt**...." (Emphasis added.) Applicants respectfully submit that **<u>disabling the output comparison</u> per Raina does <u>not</u> comprise a <u>system halt</u>**. <u>If the Examiner maintains this rejection, then Applicants respectfully request an explanation of how the cited portion of Raina discloses the limitation of claim 14.</u>

Conclusion

For the above-discussed reasons, applicants respectfully submit that each of the pending claims is patentably distinguished over the cited art. Favorable action is respectfully requested.

The Examiner is also invited to call the below-referenced attorney to discuss this case.

Respectfully Submitted,

Ken Gary Pomaranski, et al.

Dated: February 25, 2008

[illegible]

James K. Okamoto, Reg. No. 40,110
Okamoto & Benedicto LLP
P.O.Box 641330
San Jose, CA 95164-1330
Tel: (408) 436-2111
Fax: (408) 436-2114

CERTIFICATE OF MAILING			
I hereby certify that this correspondence, including the enclosures identified herein, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.			
Signature:	[illegible]		
Typed or Printed Name:	James K. Okamoto	Dated:	2/25/2008
Express Mail Mailing Number (*optional*):			